REMARKS

In the Office Action mailed on January 15, 2003, claims 17 and 41 were rejected under 35 U.S.C. § 112, second paragraph, for indefiniteness; claims 1-7, 9-13, 15, 18-37, 39, and 41-48 were rejected under 35 U.S.C. § 102(b) as being anticipated by Tran et al. (U.S. Patent No. 5,875,324) ("Tran"); and claims 8, 14, 16, 38, and 40 were objected to as being dependent upon a rejected base claim. The foregoing objections and rejections are respectfully traversed.

Claims 1-48 are pending in the subject application, of which claims 1, 17, 26, and 41 are independent.

Foreign Priority:

The Examiner has not yet acknowledged the Applicant's claim for foreign priority and submission of a certified copy of the foreign priority document filed on March 21, 2000. The Applicant respectfully requests that the Examiner acknowledge the same.

Amendments to the Claims:

Claims 8, 14, 16, 38, and 40 are amended herein to be rewritten in independent form. Claims 1, 17, 26, and 41 are amended herein to recite that "a fetch request precedes writing of the branch history information." Care has been exercised to avoid the introduction of new matter.

Objections to the Claims:

Because claims 8, 14, 16, 38, and 40 are rewritten into independent form, they are presumed to be allowable. The Applicant respectfully requests that the Examiner withdraw the objections thereto and allow the claims.

Rejections of the Claims:

Tran:

The RAM of the branch history table/branch prediction array in Tran is a 1RW type (read/write cannot be executed in parallel). If update information that is presented for update

matches a fetch request, then the update is bypassed and prediction information is forwarded, because the presented update information is just subsequent prediction information. In contrast, as described later, the present invention limits such bypass control to an RSW configuration (claim 23 of the present invention).

Tran is limited (in claim 1) to a branch prediction unit that is configured to delay updating the branch prediction array until a first clock cycle in which the fetch address is inhibited (other claims are the dependent claims thereof). Therefore, the subject matter of Tran is that array reading by a fetch request cannot be inherently interfered with by writing. In contrast, the present invention does not require the branch prediction unit to delay updating until the fetch request is prohibited. The present invention simply requires the branch prediction unit to delay updating in such a way as not to degrade the function by not receiving a fetch request.

Specifically, an operation to fixedly delay updating by several clock cycles should not always be a fetch inhibit after several clock cycles. However, because several fetch requests can be issued by delaying updating by several clock cycles, function degradation can be suppressed. Specifically, in Tran, appropriate timing is the beginning of fetch inhibit timing, while in the present invention, it is timing that does not directly affect the issuance of a fetch request (it is acceptable if a fetch request precedes updating, and it is also acceptable if updating is made in the middle of a fetch inhibit cycle). In the present invention, there is read/write interference and write priority is allowed in such a way as not to degrade the function. This is incompatible with the complete inhibition suggested by Tran.

Regarding claim 3, a fetch request is not prohibited at all, and it is acceptable if writing is made in a prefetch request cycle, i.e., in a state where a demand fetch request is precedently issued in such a way that there is no need of a demand fetch request. Whether a prefetch request conducts branch prediction depends on its configuration. Whether a prefetch request is prohibited and writing is made with priority or the issuance of a prefetch request and writing are conducted in parallel also depends on its configuration.

Regarding claim 4, Tran does not disclose or suggest the fact that delaying writing by several clock cycles prevents function degradation and enables writing.

Regarding claims 5, 6 and 7, the same comment applies to each of them.

Regarding claim 9, the same comment also applies to it, although it has a different

delaying means.

RSW in claim 10, that is, control by a plurality of update buffers indicat s control exercised in the case where a plurality of pieces of update information arrives when an update is delayed. However, Tran does not teach the configuration using a plurality of update buffers. For example, Tran describes that the microprocessor delays writing until there is a second prediction miss and it teaches that it must write either a first/second miss-predictions if there is a second miss-prediction.

Since the execution unit has no instruction while writing is delayed as described in claims 4 through 9 (because it is re-fetched, prediction execution is cancelled), it also indicates the fact that only one update buffer is sufficient because there is no more update request.

Claims 17 and 41:

Although the Examiner indicates that what branch instruction has been executed (dualism of execution) is ambiguous, the present invention mainly assumes out-of-order execution control and the term "executed" means "committed the execution" (because a pipeline process also includes such a situation, it should not be limited to out-of-order execution).

The relationship between a branch instruction and the result of its execution is as follows:

Branch instruction ---<to execute>===<to complete>...<to commit>

↓

BRHIS update request

The execution of a branch instruction means the following processes.

- It determines conditions after a condition code is confirmed.

- It determines whether branch prediction is correct, and if it is not correct, it modifies it.

When these processes finish, the branch instruction is "completed" and the microprocessor enters the state of waiting for "to commit". At this moment, an update request is issued to a branch history. "To commit" means the process of waiting for the execution process of a group of instructions to be executed logically before a branch instruction (or an exception process, an interruption process, etc.) and committing the logical execution order.

If a situation that changes an instruction sequence, such as interruption, etc., occurs, a

waiting state after then or a group of instructions in execution are all discarded (in th pr ferred embodiment, the case wh re a signal RS1 is turned on represents it).

"A branch instruction has been executed" means the state where the execution process of a single branch instruction is completed and the process waits for the termination, but it cannot terminate due to interruption, etc.

Conclusion:

Withdrawal of the foregoing objections and rejections is respectfully requested.

There being no further objections or rejections, it is submitted that the application is in condition for allowance, which action is courteously requested. Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters. If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 6-12-2003

By: _____

Matthew Q. Ammon
Registration No. 50,346

700 Eleventh Street, NW, Suite 500
Washington, D.C. 20001
(202) 434-1500